# Clustering And Dimensionality

Jason Antal

2024-08-18

The data in wine.csv contains information on 11 chemical properties of 6500 different bottles of vinho verde wine from northern Portugal. In addition, two other variables about each wine are recorded:

```
whether the wine is red or white
the quality of the wine, as judged on a 1-10 scale by a panel of certified wine snobs.
```

Run PCA, tSNE, and any clustering algorithm of your choice on the 11 chemical properties (or suitable transformations thereof) and summarize your results. Which dimensionality reduction technique makes the most sense to you for this data? Convince yourself (and me) that your chosen approach is easily capable of distinguishing the reds from the whites, using only the "unsupervised" information contained in the data on chemical properties. Does your unsupervised technique also seem capable of distinguishing the higher from the lower quality wines? Present appropriate numerical and/or visual evidence to support your conclusions.

To clarify: I'm not asking you to run a supervised learning algorithms. Rather, I'm asking you to see whether the differences in the labels (red/white and quality score) emerge naturally from applying an unsupervised technique to the chemical properties. This should be straightforward to assess using plots.

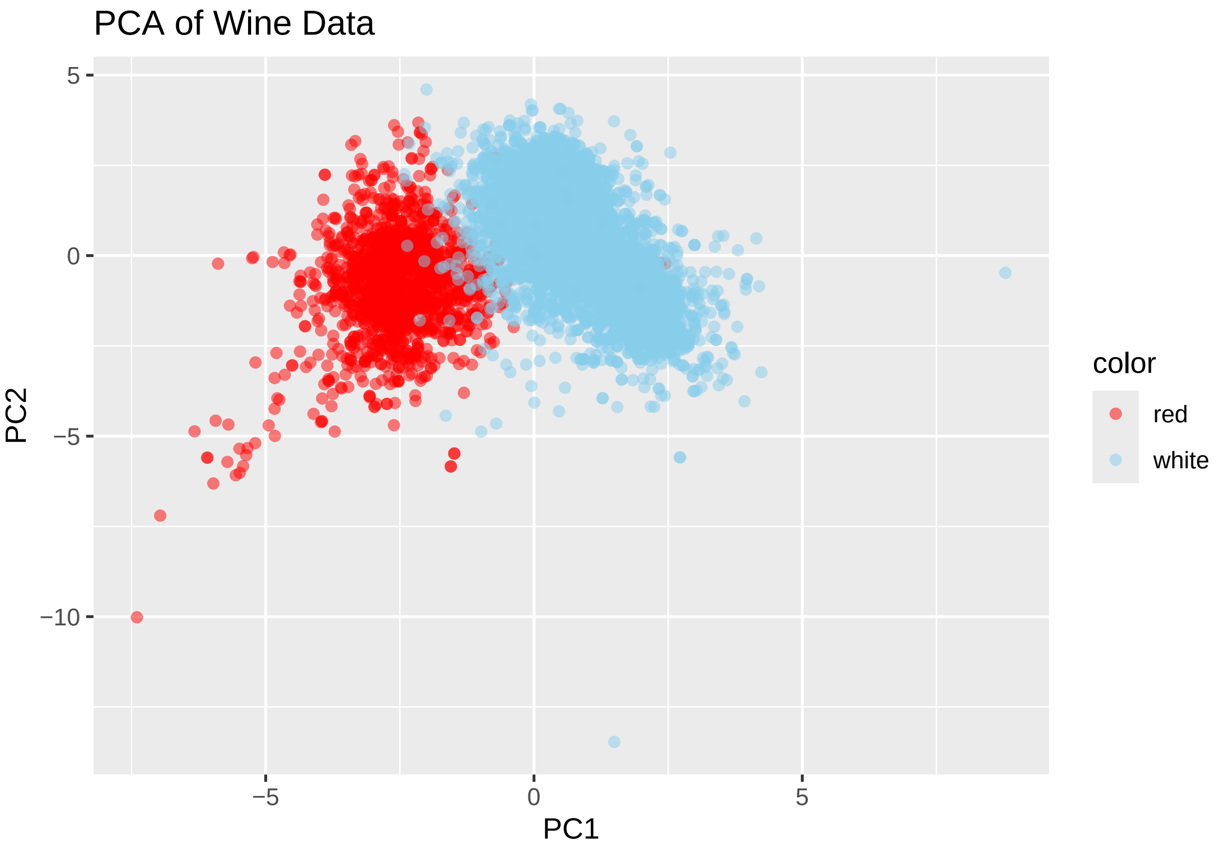
```r
# Clean data
wine_data$color <- as.factor(wine_data$color)
wine_features <- wine_data %>% select(-color, -quality)
wine_labels <- wine_data %>% select(color, quality)

cat("We're using PCA with scaling to account for different units of measurement
    in the chemical properties. We're focusing on the first two principal
    components for visualization purposes.", "\n")
```

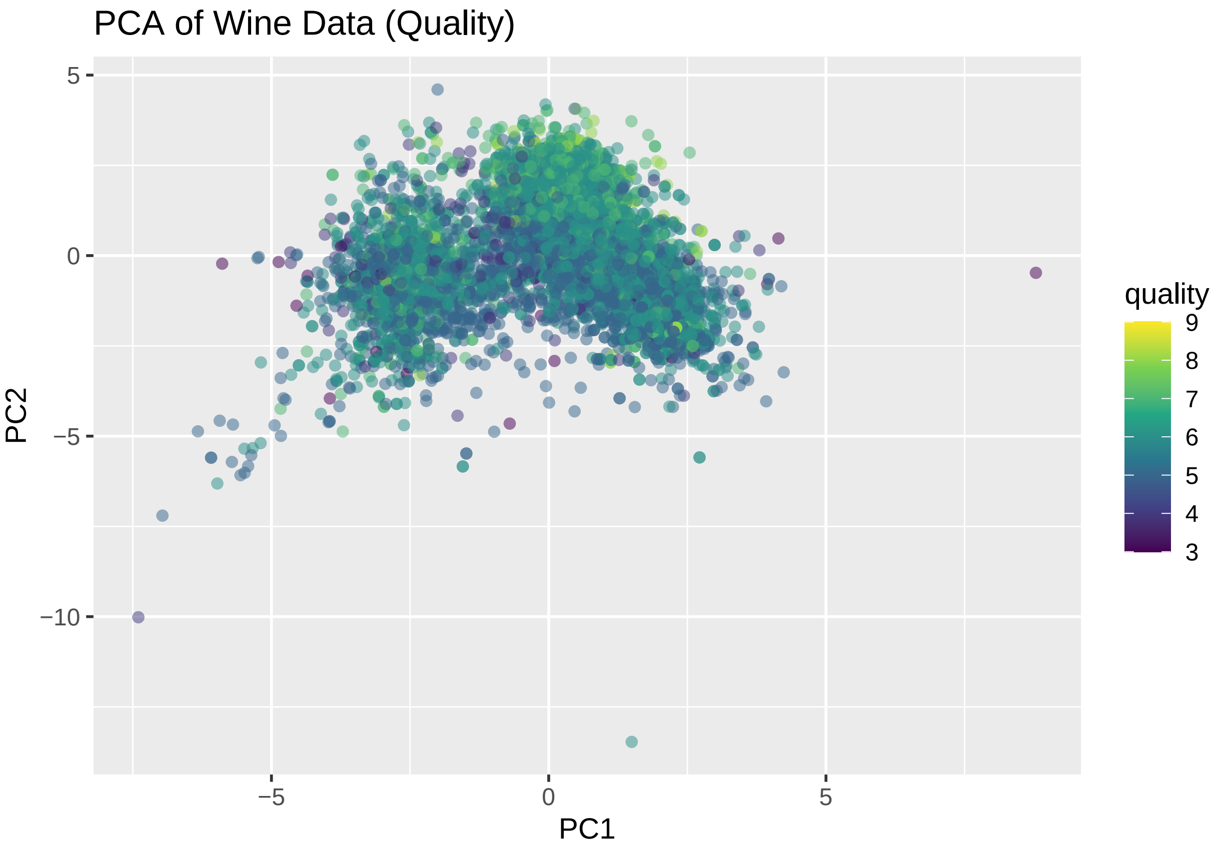
```
## We're using PCA with scaling to account for different units of measurement
##     in the chemical properties. We're focusing on the first two principal
##     components for visualization purposes.
```

```r
# Perform PCA
pca_result <- prcomp(wine_features, scale. = TRUE)
pca_df <- as.data.frame(pca_result$x[, 1:2])
pca_df$color <- wine_labels$color
pca_df$quality <- wine_labels$quality

# Plot PCA results
ggplot(pca_df, aes(x = PC1, y = PC2, color = color)) +
  geom_point(alpha = 0.5) +
  scale_color_manual(values = c("red" = "red", "white" = "skyblue")) +
  labs(title = "PCA of Wine Data", x = "PC1", y = "PC2")
```



```r
ggplot(pca_df, aes(x = PC1, y = PC2, color = quality)) +
  geom_point(alpha = 0.5) +
  scale_color_viridis_c() +
  labs(title = "PCA of Wine Data (Quality)", x = "PC1", y = "PC2")
```



```r
wine_data_unique <- unique(wine_data) #TSNE throws error if it sees duplicates
wine_features_unique <- wine_data_unique %>% select(-color, -quality)

wine_features_unique <- unique(wine_features_unique) # for some reason, unique(wine_data) does not remo

#If the below lines are not run to assign labels, a row mismatch error occurs
wine_data$id <- 1:nrow(wine_data)
features_with_id <- cbind(wine_features_unique, id = 1:nrow(wine_features_unique))
# Merge with original data to get corresponding labels
wine_data_unique <- merge(features_with_id, wine_data[, c("id", "color", "quality")], by = "id")
wine_data_unique$id <- NULL

set.seed(2)
tsne_result <- Rtsne(wine_features_unique, dims = 2, perplexity = 30, verbose = TRUE, max_iter = 200)
```

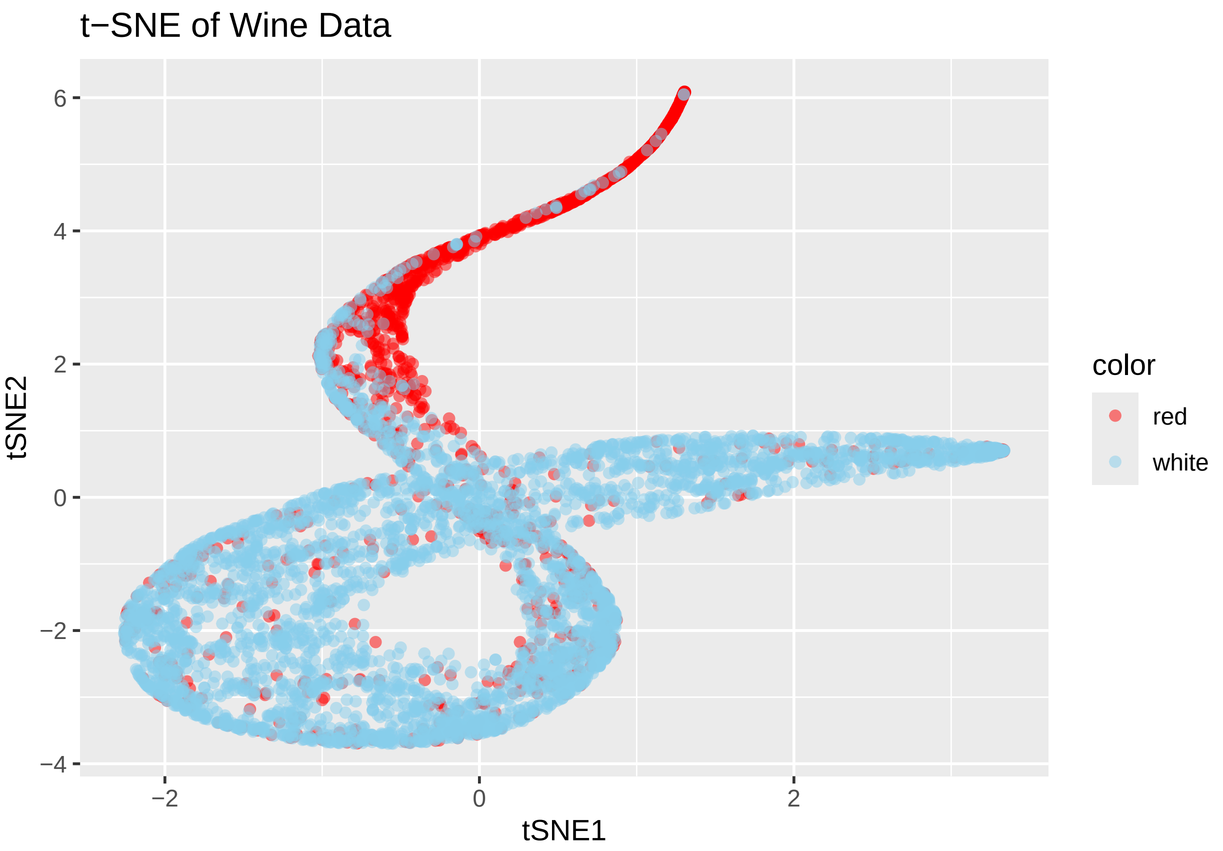
```
## Performing PCA
## Read the 5318 x 11 data matrix successfully!
## OpenMP is working. 1 threads.
## Using no_dims = 2, perplexity = 30.000000, and theta = 0.500000
## Computing input similarities...
## Building tree...
## Done in 0.96 seconds (sparsity = 0.020605)!
## Learning embedding...
## Iteration 50: error is 90.919748 (50 iterations in 0.53 seconds)
## Iteration 100: error is 73.836875 (50 iterations in 0.51 seconds)
## Iteration 150: error is 71.798221 (50 iterations in 0.66 seconds)
## Iteration 200: error is 70.929988 (50 iterations in 0.90 seconds)
## Fitting performed in 2.61 seconds.
```

```
cat("We're using tSNE with 2 dimensions for easy visualization. Perplexity
    30 is chosen as a balance between local and global structure preservation.", "\n")
```

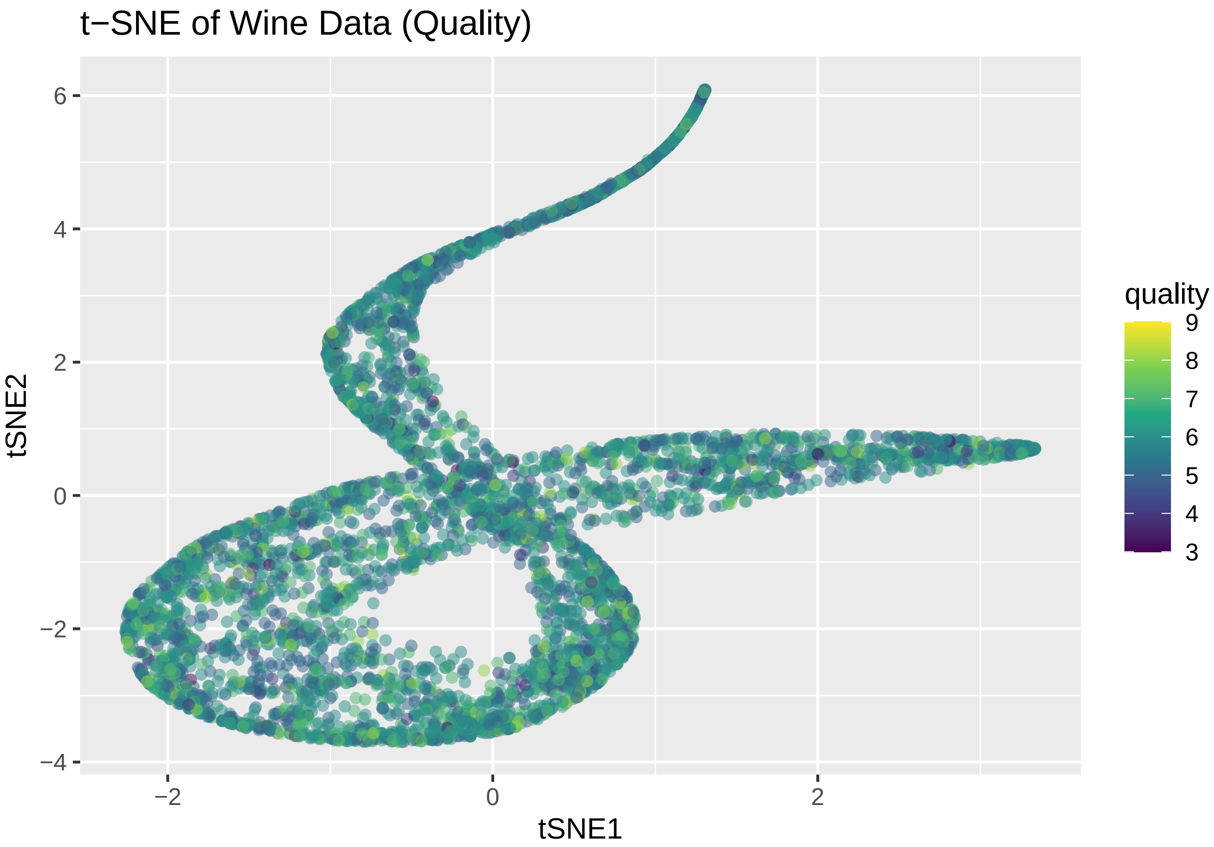
```
## We're using tSNE with 2 dimensions for easy visualization. Perplexity
##     30 is chosen as a balance between local and global structure preservation.
```

```
tsne_df <- data.frame(
  tSNE1 = tsne_result$Y[,1],
  tSNE2 = tsne_result$Y[,2],
  color = wine_data_unique$color,
  quality = wine_data_unique$quality
)

# Plot t-SNE results
ggplot(tsne_df, aes(x = tSNE1, y = tSNE2, color = color)) +
  geom_point(alpha = 0.5) +
  scale_color_manual(values = c("red" = "red", "white" = "skyblue")) +
  labs(title = "t-SNE of Wine Data", x = "tSNE1", y = "tSNE2")
```



```
ggplot(tsne_df, aes(x = tSNE1, y = tSNE2, color = quality)) +
  geom_point(alpha = 0.5) +
  scale_color_viridis_c() +
  labs(title = "t-SNE of Wine Data (Quality)", x = "tSNE1", y = "tSNE2")
```



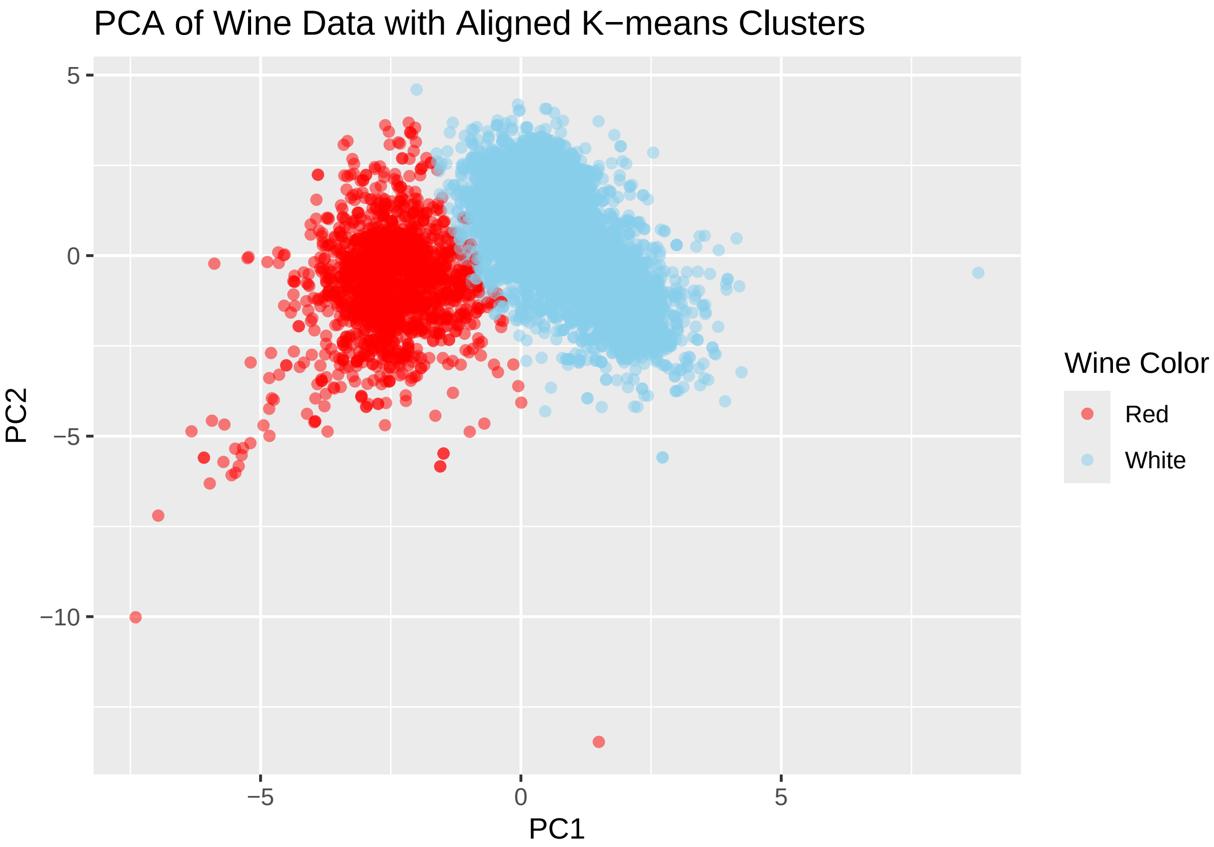
```
set.seed(2)
scaled_features <- scale(wine_features)
kmeans_result <- kmeans(scaled_features, centers = 2, nstart = 25)

align_labels <- function(true_labels, cluster_labels) {
  confusion <- table(true_labels, cluster_labels)
  if (confusion[1,1] + confusion[2,2] < confusion[1,2] + confusion[2,1]) {
    return(3 - cluster_labels)  # Swap 1 and 2
  } else {
    return(cluster_labels)
  }
}

aligned_clusters <- align_labels(wine_labels$color, kmeans_result$cluster)

pca_df$aligned_cluster <- as.factor(aligned_clusters)

ggplot(pca_df, aes(x = PC1, y = PC2, color = aligned_cluster)) +
  geom_point(alpha = 0.5) +
  labs(title = "PCA of Wine Data with Aligned K-means Clusters", x = "PC1", y = "PC2") +
  scale_color_manual(values = c("1" = "red", "2" = "skyblue"),
                     labels = c("1" = "Red", "2" = "White"),
                     name = "Wine Color")
```



```
tsne_df$aligned_cluster <- as.factor(aligned_clusters[match(1:nrow(tsne_df), match(wine_features_unique

confusion_matrix <- table(aligned_clusters, wine_labels$color)
print(confusion_matrix, sep = "")
```

```
##
## aligned_clusters  red white
##                1 1575    68
##                2   24  4830
```

```
accuracy <- sum(diag(confusion_matrix)) / sum(confusion_matrix)
cat("Accuracy:", round(accuracy * 100, 2), "%", sep = "")
```

```
## Accuracy:98.58%
```

```
cat("Results summary:", "\n",
"-t-SNE was effective for this wine dataset, offering accurate separation
between red & white wines.", "\n",
"-The t-SNE visualization shows two distinct clusters, aligning closely with wine
colors without using color labels in the computation.", "\n",
"-K-means clustering supports this, achieving 98.6% accuracy in separating red and
white wines based on chemical properties alone.", "\n",
"-However, neither t-SNE nor PCA show clear patterns corresponding to wine quality
scores, suggesting a more complex relationship between chemical properties and quality.", "\n",
"-These methods excel at identifying wine colors but struggle with judging quality. This
indicates that color is more related to chemical composition, while quality is not.", sep = "")
```

```
## Results summary:
## -t-SNE was effective for this wine dataset, offering accurate separation
## between red & white wines.
## -The t-SNE visualization shows two distinct clusters, aligning closely with wine
## colors without using color labels in the computation.
## -K-means clustering supports this, achieving 98.6% accuracy in separating red and
## white wines based on chemical properties alone.
## -However, neither t-SNE nor PCA show clear patterns corresponding to wine quality
## scores, suggesting a more complex relationship between chemical properties and quality.
## -These methods excel at identifying wine colors but struggle with judging quality. This
## indicates that color is more related to chemical composition, while quality is not.
```